Supplementary Material for
Mean-Shift Feature Transformer

Takumi Kobayashi[†‡]
[†]National Institute of Advanced Industrial Science and Technology, Japan
[‡]University of Tsukuba, Japan
takumi.kobayashi@aist.go.jp

A. WEIGHT representation

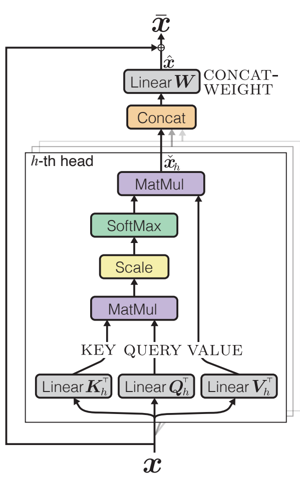
In [9], back-projection by WEIGHT is originally applied to the *concatenated* subspace feature representation (Figure Aa) as

$$\hat{\boldsymbol{x}} = \mathtt{concat}_{\rightarrow}\Big[\{\boldsymbol{W}_h\}_{h=1}^H\Big]\mathtt{concat}_{\downarrow}\Big[\{\boldsymbol{V}_h^\top \mathcal{X}\,\sigma(\mathcal{X}^\top \boldsymbol{K}_h \boldsymbol{Q}_h^\top \boldsymbol{x})\}_{h=1}^H\Big], \tag{i}$$

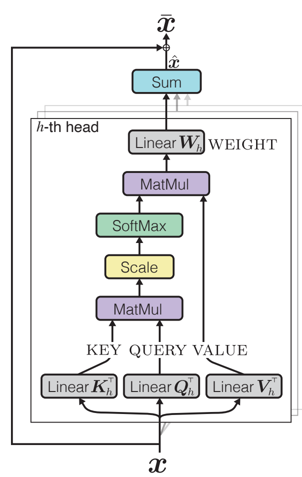
which employs linear weight of $\boldsymbol{W} = \mathtt{concat}_{\rightarrow}(\{\boldsymbol{W}_h\}_{h=1}^H) \in \mathbb{R}^{d\times H\hat{d}}$, a horizontal concatenation of multi-head WEIGHT $\{\boldsymbol{W}_h \in \mathbb{R}^{d\times\hat{d}}\}_{h=1}^H$. It should be noted that it is equivalent to our reformulation (ii) by dividing the weight $\boldsymbol{W}$ into head-wise WEIGHT which is well interpretable from the viewpoint of mean-shift updating (Section 2);

$$\hat{\boldsymbol{x}} = \sum_{h=1}^H \boldsymbol{W}_h\Big[\boldsymbol{V}_h^\top \mathcal{X}\,\sigma(\mathcal{X}^\top \boldsymbol{K}_h \boldsymbol{Q}_h^\top \boldsymbol{x})\Big]. \tag{ii}$$

Figure A depicts (superficial) difference of those formulations (i, ii) which are intrinsically identical.



(a) Original formulation [9]



(b) Our reformulation (Figure 1a)

Figure A. Architectural difference regarding WEIGHT in the original transformer formulation (a) in [9] and our reformulation (b), a.k.a Figure 1a, which are mathematically described in (i) and (ii), respectively, and are intrinsically identical.

B. Softmax with Gaussian kernel

The proposed MSF-transformer feeds a Gaussian kernel to a softmax function by

$$\sigma\left(\left\{-\frac{1}{2}\|\boldsymbol{K}_h^\top \boldsymbol{x}_i - \boldsymbol{Q}_h^\top \boldsymbol{x}\|_2^2\right\}_{i=1}^m\right) = \left\{\frac{\exp(\hat{d}^{-\frac{1}{2}}\boldsymbol{x}_i^\top \boldsymbol{K}_h \boldsymbol{Q}_h^\top \boldsymbol{x} - \frac{1}{2}\hat{d}^{-\frac{1}{2}}\|\boldsymbol{K}_h^\top \boldsymbol{x}_i\|_2^2 - \frac{1}{2}\hat{d}^{-\frac{1}{2}}\|\boldsymbol{Q}_h^\top \boldsymbol{x}\|_2^2)}{\sum_{j=1}^m \exp(\hat{d}^{-\frac{1}{2}}\boldsymbol{x}_j^\top \boldsymbol{K}_h \boldsymbol{Q}_h^\top \boldsymbol{x} - \frac{1}{2}\hat{d}^{-\frac{1}{2}}\|\boldsymbol{K}_h^\top \boldsymbol{x}_j\|_2^2 - \frac{1}{2}\hat{d}^{-\frac{1}{2}}\|\boldsymbol{Q}_h^\top \boldsymbol{x}\|_2^2)}\right\}_{i=1}^m \tag{iii}$$

$$= \left\{\frac{\exp(\hat{d}^{-\frac{1}{2}}\boldsymbol{x}_i^\top \boldsymbol{K}_h \boldsymbol{Q}_h^\top \boldsymbol{x} - \frac{1}{2}\hat{d}^{-\frac{1}{2}}\|\boldsymbol{K}_h^\top \boldsymbol{x}_i\|_2^2)}{\sum_{j=1}^m \exp(\hat{d}^{-\frac{1}{2}}\boldsymbol{x}_j^\top \boldsymbol{K}_h \boldsymbol{Q}_h^\top \boldsymbol{x} - \frac{1}{2}\hat{d}^{-\frac{1}{2}}\|\boldsymbol{K}_h^\top \boldsymbol{x}_j\|_2^2)}\right\}_{i=1}^m \tag{iv}$$

$$= \sigma\left(\left\{\boldsymbol{x}_i^\top \boldsymbol{K}_h \boldsymbol{Q}_h^\top \boldsymbol{x} - \frac{1}{2}\|\boldsymbol{K}_h^\top \boldsymbol{x}_i\|_2^2\right\}_{i=1}^m\right), \tag{v}$$

where $\|\boldsymbol{K}_h^\top \boldsymbol{x}_i\|_2^2$ are (pre-)computed independently of $\boldsymbol{x}$. Therefore, we can compute the softmax at a negligible extra cost, just for $\|\boldsymbol{K}_h^\top \boldsymbol{x}_i\|_2^2$, in comparison to the standard one $\sigma\left(\left\{\boldsymbol{x}_i^\top \boldsymbol{K}_h \boldsymbol{Q}_h^\top \boldsymbol{x}\right\}_{i=1}^m\right)$ used in the original transformer.

C. ViT architecture

In the experiments (Section 3), we employ the simplified architecture [1] of Vision Transformer (ViT). It simplifies the original ViT [2] mainly by applying (1) global average pooling (GAP) instead of using `[CLS]` token which is followed by a single linear classifier and (2) fixed 2D-sinusoidal position embedding added to patch embedding of 16×16 pixels; architectural parameters for various ViTs are shown in Table B. As reported in [1] and Section 3, so simplified ViTs produce superior performance to the original ones [2].

D. Training protocol

We train all the networks from scratch by using the training parameters shown in Table A on 4 GPUs, in which batch size B and beta-distribution parameter α in mixing augmentation vary with network sizes; strength of data augmentation is controlled by α. They work well even on shorter training epochs, e.g., 100 epochs.

Optimizer		AdamW [5]
Training epochs		T-epoch, $T \in \{100, 300\}$
Learning rate		$0.001 \searrow 0$ (cosine-schedule)
Warmup	epochs	$(0.1 \cdot T)$-epoch
	learning rate	$\frac{1}{30} \cdot 0.001 \nearrow 0.001$ (linear-schedule)
Weight decay		0.05
Batch size		B
Data augmentation		*Crop*: Random resized crop (224×224) [6]
		Appearance: Random choice of {Gray-scale, Solarization, Gaussian Blurring} [8] + Color jittering
		Mixing: Random choice of {MixUp [12], CutMix [11]} with beta distribution of shape parameter α
Label smoothing		$\epsilon = 0.1$

(a) Basic parameters.

CNN-Model	EfficientNet-B0 [7]	ResNet-50 [3]	ResNet-101	ResNeXt-50 (32×4) [10]	ResNeXt-101 (32×8)
Batch size B	1024	1024	1024	1024	768
Mixing Aug. α	0.1	0.2	0.2	0.2	0.2

ViT-Model	ViT-Ti [1]	ViT-SS	ViT-S	ViT-B	Swin-T [4]	Swin-S
Batch size B	1024	1024	1024	768	1024	768
Mixing Aug. α	0.1	0.1	0.2	0.5	0.5	0.5

(b) Batch size B and beta-distribution parameter α in mixing augmentation.

Table A. Training parameters.

E. Ablation study: layer depth

We analyze how the MSF-transformer module works across various depths (layers) in a network. We partition 12 layers of ViT-S in Table B into 6 blocks and embed MSF-transformer in a block-wise manner to report performance results in Table C. The proposed module works rather uniformly across depths, rendering performance improvement of $0.1 \sim 0.2$ points at any depths. These results show that the MSF model contributes to enhancing feature transformation in disregard of depth, which motivates us to fully embed the MSF-transformer module to all layers in the experiments (Section 3).

Model	Layer	Width d	Head H	MLP
ViT-Ti [1]	12	192	3	$4d$
ViT-SS [1]	6	384	6	$4d$
ViT-S [1]	12	384	6	$4d$
ViT-B [1]	12	768	12	$4d$

Table B. ViT architectures. Patch size is 16×16 and MLP indicates the size of hidden layer in 2-layered MLP following the transformer module (Figure A or Figure 1).

Layers						
1,2	3,4	5,6	7,8	9,10	11,12	Acc. (%)
✓	✓	✓	✓	✓	✓	79.79 (MSF)
-	✓	✓	✓	✓	✓	79.65
-	-	✓	✓	✓	✓	79.43
-	-	-	✓	✓	✓	79.45
-	-	-	-	✓	✓	79.20
-	-	-	-	-	✓	79.22
-	-	-	-	-	-	78.98 (Orig.)
✓	-	-	-	-	-	79.17
✓	✓	-	-	-	-	79.31
✓	✓	✓	-	-	-	79.36
✓	✓	✓	✓	-	-	79.52
✓	✓	✓	✓	✓	-	79.71

Table C. Performance comparison by embedding MSF-transformer module at various depths in 100-epoch trained ViT-S. The checkmark ✓ indicates that the MSF module is applied at the layers.

References

[1] Lucas Beyer, Xiaohua Zhai, and Alexander Kolesnikov. Better plain vit baselines for imagenet-1k. *arXiv*, 2205.01580, 2022. 2, 3

[2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 2

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 2

[4] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 10012–10022, 2021. 2

[5] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 2

[6] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, 2015. 2

[7] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, 2019. 2

[8] Hugo Touvron, Matthieu Cord, and Hervé Jégou. Deit iii: Revenge of the vit. In *ECCV*, 2022. 2

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017. 1

[10] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, pages 5987–5995, 2017. 2

[11] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, pages 6023–6032, 2019. 2

[12] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018. 2